

Comments on the ending keyword in the FOR-structure  
=====

The main reason to keep NEXT as the terminating word of a FOR-loop was that NEXT is an active statement that INCREMENTS the control variable:

```
FOR i:=1 TO 10 DO // initialize the vector 'a'
  a(i):=0
NEXT i
```

Here the control variable is assigned the initial value of 1; the first vector element is assigned; the NEXT statement increments 'i' by one; control proceeds to the FOR statement which tests if the loop terminates; assign to the second element etc.

The statements above can be written as an one-line FOR loop:

```
FOR i:=1 TO 10 DO a(i):=0 // initialize the vector 'a'
```

But it will never finish !!

because people will not expect that the control variable is incremented because of the missing NEXT.

Therefore we suggest this:

```
FOR i:=1 TO 10 DO a(i):=0 NEXT i // initialize the vector 'a'
```

which is very similar to the short REPEAT statement.

Thus we suggest the short FOR loop changed to the following:

```
FOR <var>:=<num.expr> TO <num.expr> DO <simple stmt> NEXT <var>
```

Or NEXT should be replaced by ENDFOR in the long FOR statement.

This implies that the syntax for the short FOR statement should be

```
FOR <var>:=<num.expr> TO <num.expr> DO <simple stmt>
```

The execution of a FOR loop should be regarded as if the simple statement was executed for successive values of <var> (a kind of parameter):

```
FOR ... DO      +-----+
                  !  stmt  !
                  +-----+
```

And the long FOR statement:

```
FOR i:=1 TO 10 DO
  a(i):=0
ENDFOR i
```

Here is ENDFOR regarded as block terminator for the block controlled by the FOR statement. All the necessary information for the iteration is contained in the opening statement.

The reason why the short FOR loop is not ended by a keyword is the same as for the short WHILE statement.

CBM COMAL 80 group, 28 feb 1983